



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/492,265	01/27/2000	Yi-Hsien Hao	0063-022004/BU1279	9668
57346 7590 08/12/2010 BRAKE HUGHES BELLERMANN LLP c/o CPA Global P.O. Box 52050 Minneapolis, MN 55402				
EXAMINER				
NG, CHRISTINE Y				
ART UNIT		PAPER NUMBER		
2464				
NOTIFICATION DATE		DELIVERY MODE		
08/12/2010		ELECTRONIC		

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

BILL@BRAKEHUGHES.COM  
uspto@brakehughes.com  
docketing@cpaglobal.com

# Office Action Summary

**Application No.**

09/492,265

**Applicant(s)**

HAO ET AL.

**Examiner**

CHRISTINE NG

**Art Unit**

2464

**Period for Reply** -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 03 June 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-60 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-60 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 27 January 2000 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
- ☐ Certified copies of the priority documents have been received.
  - ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  - ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/C2)
- 4) ☐ Interview Summary (PTO-413)
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_
- Paper No(s)/Mail Date \_\_\_\_\_

## DETAILED ACTION

### ***Claim Rejections - 35 USC § 103***

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

2. Claims 1, 3-5, 6, 8-11, 13-15, 28-30, 32-34, 47-50, 52-55 and 57-60 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku, and in further view of U.S. Patent No. 6,151,644 to Wu.

Referring to claim 1, Baumert et al disclose a memory structure (Figure 5) for use with a dual-speed Ethernet device, the memory structure comprising:

An Address Resolution Table (Figure 5, address table 38) comprising a plurality of locations, each of said locations being configured to store a packet destination address, where the Address Resolution Table is configured to:

Resolve addresses in a packet-based network switch (Figure 1).

Use a key to index one of said locations, wherein the key is the packet destination address associated with said indexed location. In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A Packet Storage Table (Figures 4 and 5, packet memory 20) configured to:

Receive a packet for storage in the packet-based network switch. Refer to Column 5, lines 27-28.

Share a preselected portion of memory with the Address Resolution Table. In Figure 5, packet memory 20 and address table 38 are connected to each other.

A mechanism configured to receive an individual packet (Figures 4 and 5, packet memory 20) wherein the mechanism is configured to perform only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet and execute a first single access in order to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet; Column 5, lines 27-55) and a second single access to access the packet destination address at the indexed location using the key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the entire packet is to be transmitted.

Wherein the memory structure implements memory arbitration for at least six types of memory accesses. Figure 2 shows the switch controller 23 primarily comprised of seven elements (MAC RCV buffer 30, MAC XMIT buffer 32, packet memory input buffer 34, packet memory output buffer 36, address table 38, RCV controller 40 and XMIT controller 42), each of which performs a different type of memory access. Refer to Column 3, line 57 to Column 4, line 21.

Wherein the memory structure facilitates full-duplex, non-blocked Ethernet switch operations at wire speed. The memory structure can simultaneously receive and

transmit data (full-duplex, non-blocked) across multiple LAN data ports such as 28 Ethernet ports comprised of 10/100 and/or 10Mbps ports. Refer to Column 1, lines 28-34; and Column 3, lines 21-28.

Baumert et al do not disclose wherein the key *is a predefined portion* of a packet destination address and that indexing is done *via at least an offset in address space*.

Kaku discloses a system wherein a controller extracts an incoming packet's destination address, matches the address with one of the addresses stored in an address lookup table, and reads a corresponding output port. The entire 48-bit destination address is not used in the address lookup table. Instead, memory locations in the address lookup table are addressed by a portion of the destination address in order to save space. For example, the 48-bit destination address is compressed to 32 bits and then the 5 least significant bits of the 32-bit compressed address (offset of 27 bits into the 32-bit destination address space to obtain the 5 least significant bits) may be used to address the address lookup table. Refer to Column 1, line 60 to Column 2, line 32. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the key *is a predefined portion* of a packet destination address and that indexing is done *via at least an offset in address space*. One would have been motivated to do so to reduce the lookup table to a smaller and more practical size.

Baumert et al also do not disclose a *single buffer per packet* mechanism.

Wu discloses a "maximum packet size" buffer manager that divides buffer memory into uniform-sized packet buffers, each of which is large enough to handle

the largest permissible packet size. Each buffer stores only one packet regardless of its size. A "maximum packet size" buffer manager employs simple record keeping logic; the records include the size of the packet stored in the buffer memory, where each packet is stored, and the order in which stored packets arrived. Simple recording keeping procedures decrease the complexity of the buffer manager and the amount of time the buffer manager needs to process incoming packets. Refer to Column 1, lines 25-57. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a *single buffer per packet* mechanism. One would have been motivated to do so to simplify record keeping procedures.

Referring to claim 3, Baumert et al disclose wherein the packet- based network switch implements an IEEE standard 802.3 communication protocol (Ethernet LAN's). Refer to Column 1, lines 19-34.

Referring to claim 4, Baumert et al disclose in Figure 1 wherein the switch (10) comprises a plurality of ports (18a-18n).

Referring to claim 5, Baumert et al disclose wherein the switch comprises at least 8 ports. As shown in Figure 1, switch 10 can have up to n ports, which includes 8 ports. Furthermore, refer to Column 11, lines 33-35.

Referring to claim 6, Baumert et al disclose in Figure 3 wherein the Address Resolution Table is implemented using an associative memory structure comprising one of an n-way associative memory (one-way associative memory: each destination address corresponds to one output port), a hash table (none), a binary search structure

(none), and a sequential search structure (none). Refer to Column 4, lines 5-8; and Column 7, lines 54-65.

Referring to claim 8, Baumert et al disclose a memory structure (Figure 5) comprising:

An Address Resolution Table (Figure 5, address table 38) comprising an associative memory structure comprising a plurality of locations, each of said locations being configured to store a packet destination address, wherein the Address Resolution Table is configured to resolve addresses in a packet-based network switch (Figure 1) and to use a key to index one of said locations within the Address Resolution Table. In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A mechanism configured to receive an individual packet for enabling only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet and execute a first single access in order to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet; Column 5, lines 27-55) and a second single access to locate the packet destination address at the indexed location using the key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the entire packet is to be transmitted, wherein the key comprises a packet destination address.

Wherein the memory structure implements a weighted priority, round-robin memory arbitration technique. The system can utilize a round-robin arbitration technique, which can include all forms of round-robin techniques. Refer to Column 7, lines 16-21; Column 9, lines 8-12 and lines 47-49; and Column 10, lines 36-42.

Baumert et al do not disclose wherein the key *is a predefined portion* of a packet destination address and that indexing is done *using an offset in address space*. Refer to the Kaku et al rejection part of claim 1.

Baumert et al also do not disclose a *single buffer per packet* mechanism. Refer to the Wu rejection part of claim 1.

Referring to claim 9, Baumert et al disclose in Figure 4 a Packet Storage Table (packet memory 20; Column 5, lines 27-28), the Packet Storage Table being adapted to receive at least one of each of a Packet Data Address (60,62) and a Packet Data Value (64). In Figure 3, a packet includes address data and user data. Refer to Column 5, lines 13-26.

Referring to claim 10, Baumert et al disclose in Figure 4 a Transmit Descriptor Table (receive session register 76), the Transmit Descriptor Table being associated with a corresponding packet-based network transmit port (ports 78a-78m), and the Transmit Descriptor Table being configured to receive a Table Descriptor Address (buffer address in descriptor 74) and a Table Descriptor Value (byte count in descriptor 74). Each of the M output queues 78a-78m are associated with a receive session register 76. Refer to Column 5, lines 52-60.



Referring to claim 11, Baumert et al disclose in Figure 5 wherein the associative memory structure comprises one of a direct-mapped/one-way associative memory structure (one-way associative memory: each destination address corresponds to one output port) and a two-way associative memory structure (none). Refer to Column 4, lines 5-8; and Column 7, lines 54-65.

Referring to claim 13, refer to the rejection of claim 8, claim 9 and claim 10.

Referring to claim 14, refer to the rejection of claim 6.

Referring to claim 15, Baumert et al disclose in Figure 4 wherein the memory block (packet memory 20) comprises a shared memory block (made up of buffers 70).

Referring to claim 28, Baumert et al disclose a packet-based switch (Figure 5) comprising:

A shared memory structure comprising an Address Resolution Table (Figure 5, address table 38) and a Packet Storage Table (Figures 4 and 5, packet memory 20), wherein the address resolution table comprises a plurality of locations, each of said locations being configured to store a packet destination address.

A key to index one of said locations within the Address Resolution Table. In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A mechanism configured to receive an individual packet, perform only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet, and execute a first single access in order to locate the packet destination address at the indexed

location using a key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the key is the packet destination address at the indexed location, and a second single access to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet; Column 5, lines 27-55), wherein the entire packet is to be transmitted.

Wherein the packet-based switch performs full-duplex, non-blocked Ethernet switch operations at wire speed. The memory structure can simultaneously receive and transmit data (full-duplex, non-blocked) across multiple LAN data ports such as 28 Ethernet ports comprised of 10/100 and/or 10Mbps ports. Refer to Column 1, lines 28-34; and Column 3, lines 21-28.

Baumert et al do not disclose wherein the key *is a predefined portion of a packet destination address and that indexing is done using an offset in address space*. Refer to the Kaku et al rejection part of claim 1.

Baumert et al also do not disclose a *single buffer per packet* mechanism. Refer to the Wu rejection part of claim 1.

Referring to claim 29, refer to the rejection of claim 3.

Referring to claim 30, refer to the rejection of claim 4.

Referring to claim 32, Baumert et al disclose a packet-based switch comprising a memory structure (Figure 5), the memory structure comprising:

An Address Resolution Table (Figure 5, address table 38) implemented using an associative memory structure, the Address Resolution Table comprising a plurality of

locations, each of said locations being configured to store a packet destination address, wherein the Address Resolution Table is configured to resolve addresses in a packet-based network switch (Figure 1) and to use a key to index one of said locations within the Address Resolution Table, wherein the key comprises the packet destination address at the indexed location. In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A Transmit Descriptor Table (receive session register 76), the Transmit Descriptor Table being associated with a corresponding packet-based network transmit port (ports 78a-78m), and the Transmit Descriptor Table being configured to receive a Table Descriptor Address (buffer address in descriptor 74) and a Table Descriptor Value (byte count in descriptor 74). Each of the M output queues 78a-78m are associated with a receive session register 76. Refer to Column 5, lines 52-60.

A Packet Storage Table (packet memory 20; Column 5, lines 27-28), the Packet Storage Table adapted to receive at least one of each of a Packet Data Address (60,62) and a Packet Data Value (64). In Figure 3, a packet includes address data and user data. Refer to Column 5, lines 13-26.

A mechanism configured to receive an individual packet for enabling only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet and execute a first single access in order to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet;

Column 5, lines 27-55) and a second single access to locate the packet destination address at the indexed location using the key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the entire packet is to be transmitted.

Wherein the memory structure facilitates full-duplex, non-blocked Ethernet switch operations at wire speed. The memory structure can simultaneously receive and transmit data (full-duplex, non-blocked) across multiple LAN data ports such as 28 Ethernet ports comprised of 10/100 and/or 10Mbps ports. Refer to Column 1, lines 28-34; and Column 3, lines 21-28.

Baumert et al do not disclose wherein the key *is a predefined portion* of a packet destination address and that indexing is done *using at least an offset in address space*. Refer to the Kaku et al rejection part of claim 1.

Baumert et al also do not disclose a *single buffer per packet* mechanism. Refer to the Wu rejection part of claim 1.

Referring to claim 33, refer to the rejection of claim 6.

Referring to claim 34, refer to the rejection of claim 15.

Referring to claim 47, refer to the rejection of claim 3.

Referring to claim 48, refer to the rejection of claim 4.

Referring to claim 49, Baumert et al disclose in Figure 1 wherein the switch comprises at least 4 ports. In Figure 1, switch 10 can have up to n ports, which includes 4 ports.

Referring to claim 50, refer to the rejection of claim 5.

Referring to claim 52, Baumert et la disclose a packet-based switch comprising an Address Resolution Table (Figure 5, address table 38) comprising a one-way associative memory structure (refer to the rejection of claim 6) comprising a plurality of locations, each of said locations being configured to store a packet destination address, wherein the Address Resolution Table is configured to use a key to index one of said locations within the Address Resolution Table and a Packet Data Buffer Table (Figures 4 and 5, packet memory 20) configured to share a memory block with an Address Resolution Table (Figure 5, packet memory 20 and address table 38 are connected to each other). In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A mechanism configured to receive an individual packet for enabling only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet and execute a first single access in order to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet; Column 5, lines 27-55) and a second single access to locate the packet destination address at the indexed location using the key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the key is the located packet destination address, wherein the entire packet is to be transmitted.

Wherein the memory structure facilitates full-duplex, non-blocked Ethernet switch operations at wire speed. The memory structure can simultaneously receive and transmit data (full-duplex, non-blocked) across multiple LAN data ports such as 28 Ethernet ports comprised of 10/100 and/or 10Mbps ports. Refer to Column 1, lines 28-34; and Column 3, lines 21-28.

Baumert et al do not disclose wherein the key *is a predefined portion* of a packet destination address and that indexing is done *using at least an offset in address space*. Refer to the Kaku et al rejection part of claim 1.

Baumert et al also do not disclose a *single buffer per packet* mechanism. Refer to the Wu rejection part of claim 1.

Referring to claim 53, refer to the rejection of claim 4.

Referring to claim 54, refer to the rejection of claim 49.

Referring to claim 55, refer to the rejection of claim 5.

Referring to claim 57, Baumert et al disclose a packet-based switch comprising an Address Resolution Table (Figure 5, address table 38) implemented using a direct-mapped/one-way associative memory structure (refer to the rejection of claim 6) comprising a plurality of locations, each of said locations being configured to store a packet destination address, wherein the Address Resolution Table is configured to resolve addresses in a packet-based network switch, and to use a key in index a one of said locations. In Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet. Refer to Column 4, lines 5-8; Column 5, lines 13-18; and Column 7, lines 54-65.

A mechanism configured to receive an individual packet, to perform only one transmit descriptor (Figure 4, descriptor 74) read per said individual packet, and to execute a first single access in order to locate the entire packet (in Figure 4, the buffer address pointer in descriptor 74 points to the buffer 70 that is used to store a particular packet; Column 5, lines 27-55) and a second single access to locate the packet destination address at the indexed location using the key (in Figure 5, the address table state machine 100 uses the address table 38 to determine the destination port according to the destination address of a packet; Column 7, lines 54-65), wherein the key is a located packet destination address, wherein the entire packet is to be transmitted.

Wherein the packet-based switch makes a transmit descriptor request during a transmission of a previous frame. As shown in Figure 8, a transmit descriptor request 74 is made in a previous frame because after the request is made, descriptors 74 are fetched from the port queue 78 in the descriptor memory 22 and stored in transmit session register 156 (step 166). Afterwards, the packet can be transmitted in the next frame (steps 168-188). Refer to Column 9, line 63 to Column 10, line 34.

Wherein the memory structure facilitates full-duplex, non-blocked Ethernet switch operations at wire speed. The memory structure can simultaneously receive and transmit data (full-duplex, non-blocked) across multiple LAN data ports such as 28 Ethernet ports comprised of 10/100 and/or 10Mbps ports. Refer to Column 1, lines 28-34; and Column 3, lines 21-28.

Baumert et al do not disclose wherein the key *is a predefined portion* of a packet destination address and that indexing is done *using at least an offset in address space*. Refer to the Kaku et al rejection part of claim 1.

Baumert et al also do not disclose a *single buffer per packet* mechanism. Refer to the Wu rejection part of claim 1.

Referring to claim 58, Baumert et al disclose in Figure 4 wherein the direct-mapped/one-way associative memory is searched using a destination address key direct-mapped address search. Each destination address corresponds to one output port. Refer to Column 4, lines 5-8; and Column 7, lines 54-65.

Referring to claim 59, refer to the rejection of claim 3.

Referring to claim 60, refer to the rejection of claim 4.

3. Claims 2, 18-21, 23-25, 37-40 and 42-44 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu, and in further view of U.S. Patent No. 6,021,132 to Muller et al.

Referring to claim 2, Baumert et al disclose in Figure 4 a Transmit Descriptor Table (receive session registers 76) being associated with a corresponding packet-based network transmit port (ports 78a-78m). Each of the M output queues 78a-78m are associated with a receive session register 76. Refer to Column 5, lines 52-60.

Baumert et al do not disclose a Free Buffer Pool comprising a plurality of memory buffers, each of the plurality of memory buffers having a pre-determined number of memory locations associated therewith.



Muller et al disclose in Figures 3A and 3B a shared memory manager 220 that manages the shared pool of free buffers in shared memory 230. Shared memory manager 220 returns buffers to the free pool once an output port 206 has finished transmitting packet data from the buffer. Each buffer may be of the same size with a predetermined number of memory lines for storing packet data. Refer to Column 7, lines 20-28 and Column 8, lines 22-51. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a Free Buffer Pool comprising a plurality of memory buffers, each of the plurality of memory buffers having a pre-determined number of memory locations associated therewith. One would have been motivated to do so to allow for dynamic buffer allocation and re-use of free buffers.

Referring to claims 18 and 37, refer to the Muller et al rejection part of claim 2.

Referring to claims 19 and 38, Baumert et al do not disclose wherein the Free Buffer Pool further comprises a buffer control memory.

Muller et al disclose in Figures 3A and 3B a shared memory manager 220 that manages the shared pool of free buffers in shared memory 230. Shared memory manager 220 returns buffers to the free pool once an output port 206 has finished transmitting packet data from the buffer. Refer to Column 7, lines 20-28 and Column 8, lines 22-51. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the Free Buffer Pool further comprises a buffer control memory. One would have been motivated to do so in order to provide a means to control dynamic buffer allocation.

Referring to claims 20 and 39, Baumert et al do not disclose wherein the free buffer pool control memory comprises a plurality of data bits, ones of the plurality of data bits uniquely corresponding to ones of the pre-determined number of memory locations.

Muller et al disclose in Figures 3A and 3B a shared memory manager 220 that has a count array 430 for managing the shared pool of free buffers in shared memory 230. The location of a given count field in the count array 430 represents the start address of the corresponding buffer in the shared memory 230. Also, the shared memory 230 is made up of a plurality of buffers; each of the buffers may be of the same size. Refer to Column 8, lines 37-51; and Column 9, line 65 to Column 10, line 7. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the free buffer pool control memory comprises a plurality of data bits, ones of the plurality of data bits uniquely corresponding to ones of the pre-determined number of memory locations. One would have been motivated to do so in order to locate and address of the buffers in the shared memory.

Referring to claims 21 and 40, Baumert et al disclose in Figures 4 and 5 wherein at least two of the Address Resolution Table, the Transmit Descriptor Table, the Packet Storage Table, and the Free Buffer Pool share a memory block (Figure 5). Address Tables 38, descriptors 74 and packet memory 20 are located within switch 10.

Referring to claims 23 and 42, Baumert et al do not disclose a free buffer manager, including: a buffer bus controller; a buffer bus register; a buffer control finite state machine operably coupled with the bus controller and the bus register; and a

buffer search engine operably coupled with the bus controller, the bus register, and the finite state machine.

Muller et al disclose in Figures 3B and 4 a free buffer manager (shared memory manager 220), including: a buffer bus controller (pointer RAM 320); a buffer bus register (arbiter 470); a buffer control finite state machine (array controller 450), operably coupled with the bus controller and the bus register; and a buffer search engine (address/data generator 460), operably coupled with the bus controller, bus register, and finite state machine. Pointer RAM 320 controls the buffers by storing usage counts for buffers, arbiter 470 arbitrates among the ports to provide only a single port with access to the pointer RAM 320, array controller 450 schedules read and write operations for the pointer RAM 320, and address/data generator 460 provides addressing into pointer RAM 320. Refer to Column 9, line 5 to Column 10, line 45. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include a free buffer manager, including: a buffer bus controller; a buffer bus register; a buffer control finite state machine operably coupled with the bus controller and the bus register; and a buffer search engine operably coupled with the bus controller, the bus register, and the finite state machine. One would have been motivated to do so to provide means to control the shared buffer for dynamic buffer allocation.

Referring to claims 24 and 43, Baumert et al do not disclose wherein the buffer bus controller comprises: a buffer free bus controller for detecting a buffer request and presenting the request to at least one of the finite state machine and the buffer search

engine; and a buffer grant bus controller for granting an available free buffer, as indicated by the buffer, as indicated by the buffer bus register.

Muller et al disclose in Figures 3B and 4 wherein the buffer bus controller (pointer RAM 320) comprises: a buffer free bus controller (count array 430) for detecting a buffer request and presenting the request to at least one of the finite state machine and the buffer search engine; and a buffer grant bus controller (tag array 420) for granting an available free buffer, as indicated by the buffer, as indicated by the buffer bus register. Count array 420 stores a count representing ports that are currently using a corresponding buffer in the shared memory 230, and tag array 420 indicates the availability of buffers that can be granted for use. Refer to Column 9, line 65 to Column 10, line 20. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the buffer bus controller comprises: -a buffer free bus controller for detecting a buffer request and presenting the request to at least one of the finite state machine and the buffer search engine; and a buffer grant bus controller for granting an available free buffer, as indicated by the buffer, as indicated by the buffer bus register. One would have been motivated to do so to provide means to control the request and granting of free buffers.

Referring to claims 25 and 44, Baumert et al do not disclose wherein the buffer search engine comprises a pipelined buffer search engine.

Muller et al discloses in Figures 3B and 4 that the buffer search engine (address/data generator 460) comprises a pipelined buffer search engine since it generates signals for the memories of the pointer RAM 320 to modify the count and tag

fields, which reflect the buffer ownership. Refer to Column 10, lines 28-35. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the buffer search engine comprises a pipelined buffer search engine. One would have been motivated to do so in order to provide a means to control dynamic buffer allocation.

4. Claims 7, 12, 31, 41 and 56 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu, and in further view of U.S. Patent No. 5,765,036 to Lim.

Referring to claims 7, 12 and 31, Baumert et al do not disclose wherein the number of memory accesses required per Ethernet frame is one of: one cycle per frame for address resolution; one cycle per frame for address learning; one cycle per frame for transmission read; one cycle per frame for transmission write; one cycle per eight bytes for a frame data read; and one cycle per eight bytes for a frame data write.

Lim discloses a shared memory device which allows for one system to write data to a memory location in one cycle and another system to read the data on the next data transfer cycle. This allows for maximum flexibility in a system using a shared memory. Refer to Column 8, lines 22-39. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the number of memory accesses required per Ethernet frame is one of: one cycle per frame for address resolution (none); one cycle per frame for address learning (none); one cycle per frame for transmission read; one cycle per frame for transmission write; one

cycle per eight bytes for a frame data read (none); and one cycle per eight bytes for a frame data write (none). One would have been motivated to do so to write data in one cycle and read data in the following cycle, thereby allowing for fast data transfer.

Referring to claims 41 and 56, Baumert do not disclose wherein a packet-based network switch including the memory structure implements an IEEE Standard 802.3 communication protocol. Refer to the rejection of claim 3.

Baumert et al also do not disclose wherein the number of memory accesses required per Ethernet frame is one of: one cycle per frame for address resolution; one cycle per frame for address learning; one cycle per frame for transmission read; one cycle per frame for transmission write; one cycle per eight bytes for a frame data read; and one cycle per eight bytes for a frame data write. Refer to the Lim et al rejection part of claims 7, 12 and 31.

5. Claims 16, 17, 35 and 36 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu, and in further view of U.S. Patent No. 6,088,793 to Liu et al.

Baumert et al do not disclose wherein the Transmit Descriptor Table comprises a FIFO memory structure (claims 16 and 35), and wherein the FIFO memory structure comprises a circular FIFO memory structure, the FIFO memory structure comprising a head memory pointer and a tail memory pointer (claims 17 and 36).

Liu et al disclose in Figure 4 line address buffer LAB 230 which is a circular FIFO buffer with 16-20 entries, a head pointer and a tail pointer. Entries are deallocated by

indexing the tail pointer of the FIFO so the first arriving packets are transmitted first.

Refer to Column 4, line 55 to Column 6, line 3. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the Transmit Descriptor Table comprises a FIFO memory structure (claims 16 and 35), and wherein the FIFO memory structure comprises a circular FIFO memory structure, the FIFO memory structure comprising a head memory pointer and a tail memory pointer (claims 17 and 36). One would have been motivated to do so in order to allow the first arriving packet in the memory structure to be serviced first and to utilize the pointers to determine packet service order in the memory structure.

6. Claim 22 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu in view of U.S. Patent No. 6,021,132 to Muller et al, and in further view of U.S. Patent No. 5,765,036 to Lim. Refer to the rejection of claims 41 and 56.

7. Claims 26, 27, 45 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu in view of U.S. Patent No. 6,021,132 to Muller et al, and in further view of U.S. Patent No. 5,940,375 to Soumiya et al.

Referring to claims 26 and 45, Baumert et al do not disclose wherein the buffer bus register comprises a LIFO memory structure.

Soumiya et al disclose in Figure 4 a system wherein quality class queues 409 are inside a common buffer 406 that make up a LIFO buffer. Refer to Column 18, lines 44-52. Therefore, it would have been obvious to one of ordinary skill in the art at the time the invention was made to include wherein the buffer bus register comprises a LIFO memory structure. One would have been motivated to do so in order to allow the top of the buffer to be serviced first.

Referring to claims 27 and 46, Baumert et al do not disclose wherein the LIFO comprises an eight-location LIFO memory structure. However, the LIFO disclosed in Soumiya et al can contain any number of locations to store packets, including eight.

8. Claim 51 is rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 6,067,300 to Baumert et al in view of U.S. Patent No. 6,279,097 to Kaku in view of U.S. Patent No. 6,151,644 to Wu in view of U.S. Patent No. 6,021,132 to Muller et al in view of U.S. Patent No. 5,940,375 to Soumiya et al, and in further view of U.S. Patent No. 5,765,036 to Lim. Refer to the rejection of claims 41 and 56.

### ***Response to Arguments***

9. Applicant's arguments filed June 3, 2010 have been fully considered but they are not persuasive.

Referring to the argument that Baumert et al do not disclose that the packet memory 20 and address table 38 share a preselected portion of memory (page 20, line 22 to page 21, line 14): As shown in Figure 5, packet memory 20 and address table 38 communicate via address table state machine 100, main receive state machine 92 and packet memory output buffer 32. The combination of the address table state machine



100, main receive state machine 92 and packet memory output buffer 32 reads on the "preselected portion of memory" since packet memory 20 and address table 38 share these components and utilize these components to communicate.

Referring to the argument that Baumert et al cannot disclose the single buffer per packet mechanism since Baumert et al disclose storing packets across multiple buffers (page 21, line 15 to page 22, line 11 and page 22, line 28 to page 23, line 7): Although Baumert et al disclose storing packet across multiple buffers, Baumert et al disclose that the size of the each buffer can be selected and is variable. If the size of the buffer is selected to be 1518 bytes, which is the maximum size of the packets, then the system of Baumert et al will be a "single buffer per packet" system, which would apply to the claimed limitations. The size of the buffer can be changed to accommodate one packet. Baumert et al also disclose that a data packet has a minimum length of 64 bytes and a case of where the size of each buffer is selected to be 64 bytes. So, if a data packet of length 64 bytes is received, the whole packet can be stored in one buffer, which reads of a "single buffer per packet" system and can apply to the claimed limitations. Refer to Column 5, lines 13-36. Furthermore, Wu is used to disclose the advantages of the use of a single buffer per packet mechanism.

Referring to the argument that Kaku do not disclose wherein the key is a predefined portion of a packet destination address and that indexing is done via at least an offset in address space (page 22, lines 12-27): Kaku discloses a system wherein a controller extracts an incoming packet's destination address, matches the address with one of the addresses stored in an address lookup table, and reads a corresponding

output port. The entire 48-bit destination address is not used in the address lookup table. Instead, memory locations in the address lookup table are addressed by a portion of the destination address in order to save space. For example, the 48-bit destination address is compressed to 32 bits and then the 5 least significant bits of the 32-bit compressed address (offset of 27 bits into the 32-bit destination address space to obtain the 5 least significant bits) may be used to address the address lookup table. Refer to Column 1, line 60 to Column 2, line 32.

### ***Conclusion***

10. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTINE NG whose telephone number is (571)272-3124. The examiner can normally be reached on M-F; 8:00 am - 5:00 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Ricky Ngo can be reached on (571) 272-3139. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

C. Ng  
July 27, 2010

/Ricky Ngo/  
Supervisory Patent Examiner, Art Unit 2464